## Amendments to the Claims

This listing of claims will replace all prior versions and listings of claims in the application.

1. (Previously Presented) A system for protecting data, comprising:

a memory in which encrypted data is stored; and

a processor coupled to the memory, the processor comprising a decryptor that decrypts the encrypted data, the decryptor being adapted to variably bit roll the encrypted data based on at least a data address, to fixedly bit shuffle the bit-rolled data, to add a first key to the bit-shuffled data and to process the added data with a second key,

wherein the processor receives an original key and the data address,

wherein the processor generates multiplexer selection bits and the first key that is a shifted version of the original key based on the original key and data address,

wherein the decryptor is adapted to variably bit roll the encrypted data by rotating bits within particular roll regions of encrypted data based on the multiplexer selection bits.

2. (Original) The system according to claim 1, wherein the decryptor is adapted to perform a single pipeline stage decryption.

3. (Previously Presented) The system according to claim 1, wherein the decryptor comprises a bit roller that rotates data in one or more roll regions of the incoming data based on the data address related to the received encrypted data and a key related to the first key.

4. (Original) The system according to claim 3, wherein the key comprises a shifted version of the first key.

5. (Original) The system according to claim 3, wherein the bit roller comprises a

plurality of multiplexers.

6. (Original) The system according to claim 5,

wherein each multiplexer comprises a multiplexer selection input,

wherein multiplexer selection bits are input at the multiplexer selection input, and

wherein the multiplexer selection bits are generated based on the address related to the received encrypted data and the key related to the first key.

7. (Original) The system according to claim 1, wherein the decryptor comprises a fixed bit shuffler.

8. (Original) The system according to claim 7, wherein the fixed bit shuffler comprises a fixed, hard-coded bit shuffler.

9. (Original) The system according to claim 7, wherein the fixed bit shuffler does not add a gate delay to the decryptor.

10. (Original) The system according to claim 1, wherein the decryptor comprises one or more two-bit adders.

11. (Original) The system according to claim 10, wherein each two-bit adder comprises three exclusive OR (XOR) gates and an AND gate.

12. (Original) The system according to claim 1, wherein the decryptor comprises an XOR block.

13. (Original) The system according to claim 12, wherein the XOR block comprises one or more XOR gates.

14. (Original) The system according to claim 13, wherein each XOR gate comprises a first input and a second input, the first input receiving a bit of the second key, the second input receiving a bit of the added data.

15. (Original) The system according to claim 1, wherein the first key is a shifted version of a key.

16. (Previously Presented) The system according to claim 15, wherein an amount of shift in the first key is based on the data address related to the received encrypted data.

17. (Original) The system according to claim 15, wherein the first key is generated substantially in parallel with the decrypting of the encrypted data.

18. (Original) The system according to claim 1, wherein the decryptor does not add a latency to a processor pipeline.

19. (Original) The system according to claim 1, wherein the decryptor does not add enough gate delays to exceed a clock cycle budget of the processor.

20. (Original) The system according to claim 1, wherein the decryptor decrypts a word of the encrypted data in a single cycle.

21. (Original) The system according to claim 1, wherein the word comprises a 64-bit

word.

22. (Original) The system according to claim 1, wherein the decryptor is adapted to receive encrypted data from the memory.

23. (Withdrawn) A system for protecting data, comprising:

a memory in which encrypted data is stored, the encrypted data being split into a plurality of roll regions of variable length, each roll region being characterized by a roll skip, a roll region length and a roll amount, the roll skip, the roll region length and the roll amount being set by at least a portion of the key that varies with a data address; and

a processor coupled to the memory, the processor comprising a decryptor that decrypts the encrypted data without adding a latency to a processor pipeline,

wherein the decryptor receives the key and the data address,

wherein decryptor comprises a variable bit roller that variably bit rolls encrypted data based on at least the key and the data address, and

wherein the decryptor decrypts a word of the encrypted data in a single cycle.

24. (Withdrawn) The system according to claim 23, wherein the decryptor decrypts the encrypted data without adding enough gate delays to exceed a clock cycle budget of the processor.

25. (Withdrawn) The system according to claim 23, wherein the decryptor ~~that~~ decrypts the encrypted data and decrypts a word of the encrypted data in a single cycle.

26. (Withdrawn) A system for securing data, comprising:

a processor that decrypts encrypted data, the processor being adapted to variably bit roll encrypted data based on at least a data address and to fixedly bit shuffle the bit-rolled data.

27. (Withdrawn) The system according to claim 26, wherein the processor is adapted to perform a single pipeline stage decryption.

28. (Withdrawn) A system according to claim 26, wherein the processor is adapted to add a first key to the bit-shuffled data and to process the added data with a second key.

29. (Withdrawn) The system according to claim 26, wherein the processor is adapted to decrypt the encrypted data without adding a latency to a processor pipeline.

30. (Withdrawn) A method for securing processor instructions, comprising:

partitioning data information into a plurality of roll regions, the roll regions being of variable length, each roll region being characterized by a roll skip, a roll region length and a roll amount, wherein the roll skip, the roll region length and the roll amount are set through bits of a portion of a first key, wherein the bits of the portion of the first key used to set the roll region length, the roll amount and the roll skip are based on an address;

variably rolling the data information based on the first key and the address related to the data information; and

hard-coded shuffling of the rolled data information; and

using one or more keys to process the data information.

31. (Withdrawn) The method according to claim 30, wherein the rolling, the shuffling and the using are part of a single pipeline stage decryption.

32. (Withdrawn) The method according to claim 30, wherein using one or more keys to

process the data information comprises adding the hard-coded data information and a shifted version of the first key.

33. (Withdrawn) The method according to claim 32, wherein using one or more keys to process the data information comprises processing the added data information with a second key using exclusive OR (XOR) gates.

34. (Withdrawn) The method according to claim 33, wherein the first key is not a function of the second key.

35. (Withdrawn) The method according to claim 30, wherein the data information comprises encrypted data information.

36. (Withdrawn) The method according to claim 30,
wherein the encrypted data information is stored in a memory, and
wherein the stored data information is accessed by a processor.

37. (Withdrawn) The method according to claim 30, wherein the rolling comprises rotating bits within one or more rolling regions of the data information.

38. (Previously Presented) The system according to claim 1,
wherein memory and the processor are part of a set top box,
wherein the memory comprises a flash memory and an SDRAM,
wherein instructions are stored in the flash memory before being moved to the SDRAM for execution by the processor, and
wherein the instructions stored in the flash memory are compressed before being moved

to the SDRAM for execution by the processor.

39. (Previously Presented) The system according to claim 1, wherein the processor uses a single pipeline stage decryption algorithm.

40. (Previously Presented) The system according to claim 1, wherein encrypted data stored in the memory has been encrypted using an encryption algorithm that varies periodically at address multiples such that repeated instructions are not encoded in the same way each time.

41. (Previously Presented) The system according to claim 1,

wherein the encrypted data stored in the memory is encrypted in a single clock cycle encryption scheme, and

wherein the processor decrypts the encrypted data in a single clock cycle decryption scheme.

42. (Previously Presented) The system according to claim 1,

wherein the memory and the processor are part of a set top box, and

wherein the processor that fixedly bit shuffles the bit-rolled data is configured as a fixed, hard-coded bit shuffler in which the fixed, hard-coded bit shuffling differs according to a class of the set top box such that different classes of set top boxes differ in their fixed, hard-coded bit shuffling.

43. (Previously Presented) The system according to claim 1,

wherein the memory and the processor are part of a device, and

wherein the processor that fixedly bit shuffles the bit-rolled data is configured as a fixed, hard-coded bit shuffler in which the fixed, hard-coded bit shuffling differs according to device class such that different classes of devices differ in their fixed, hard-coded bit shuffling.

44. (Previously Presented) The system according to claim 1,

wherein the decryptor comprises a series of two-bit adders that process incoming data bits, and

wherein values input to the series of two-bit adders relate to processing of first key and the data address.

45. (Previously Presented) The system according to claim 1,

wherein the decryptor comprises a bit swapper and a bit roller,

wherein the bit swapper is configured to provide fixed, hard-coded bit shuffling,

wherein the bit roller is configured to provide variable bit rolling,

wherein the decryptor comprises a plurality of two-bit adders,

wherein each two-bit adder receives two bits from a bit swapper that received two bits from the bit roller, and

wherein each two-bit adder receives two bits of the first key.

46. (Previously Presented) The system according to claim 1,

wherein a particular two-bit adder of the plurality of two-bit adders receives a different two bits of the first key based on different data addresses received by the processor.

47. (Previously Presented) The system according to claim 1,

wherein each two-bit adder outputs two bits that are received in an XOR block, and

wherein the XOR block receives two bits of the second key.

48. (Previously Presented) The system according to claim 47, wherein an output of the

XOR block is decrypted data.

49. (Previously Presented) The system according to claim 48, wherein the decrypted data is stored in an internal memory of the processor.

50. (Previously Presented) The system according to claim 1, wherein the second key is unrelated to the first key.

51. (Previously Presented) The system according to claim 1, wherein a portion of the first key

52. (Withdrawn) The system according to claim 23, wherein the key is shifted after a particular number of data addresses.

53. (Withdrawn) The system according to claim 23, wherein the key is shifted as a function of at least the key and the data address.

54. (Previously Presented) The system according to claim 50,

wherein the decryptor comprises a bit swapper that swaps bits output from the variable bit roller, and

wherein the decryptor comprises an adder that adds the shifted key to bits output from the bit swapper.

55. (Previously Presented) The system according to claim 50,

wherein the decryptor comprises an XOR block that processes bits output from the adder and bits from a hidden key that is unrelated to the shifted key, and wherein bits output from the XOR block are decrypted.

56. (Previously Presented) The system according to claim 1,

wherein the encrypted data is partitioned into a plurality of roll regions, the roll regions being of variable length,

wherein each roll region is characterized by a roll skip, a roll region length and a roll amount,

wherein the roll skip, the roll region length and the roll amount are set through bits of a portion of the original key, and

wherein the bits of the portion of the original key, selected based on the data address, are used to set the roll skip, the roll region length and the roll amount.

57. (Previously Presented) The system according to claim 1, wherein the bits of the portion of the original key used to set the roll skip, the roll region length and the roll amount are set using the bits of the portion of the original key which change as the data address changes.